Lec: 3

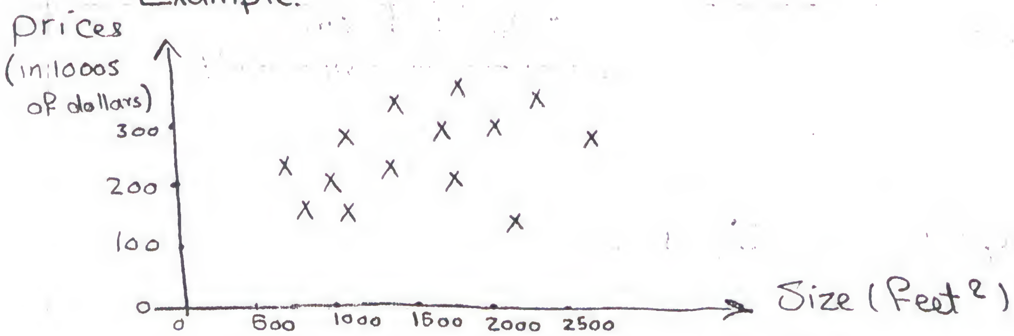
* Supervised learning
  - [I] Regression : continuous valued output.
  - Linear regression with one variable.

→ we Know that :

1- Supervised Learning : given the "right answer" for each example in the data.

2- Regression Problem : Predict real-valued output

Example.



Size (X)	Price (Y)
2104	460
1416	232
1534	315
⋮	⋮

Training Set Of housing prices.

Notation:

m = Number of training examples.

X's = "input" Variable / Features.

Y's = "output" Variable / "target" Variable.

(X, Y) is one training example

$(x^{(i)}, y^{(i)})$ is the i^{th} training example.

- we have Features (X) and right answer (Y),
- We want to make a model expresses the relationship between X and Y or we want to fit data to be able to find the value of Y at any value of X
- This model can be straight line or a curve or function of 3^{rd} degree.
- We use straight line to fit data.

Straight Line Function $y = ax + b$

Training Set

↓

Learning Algorithm

Size of house X → h → Estimated price / Estimated value of Y

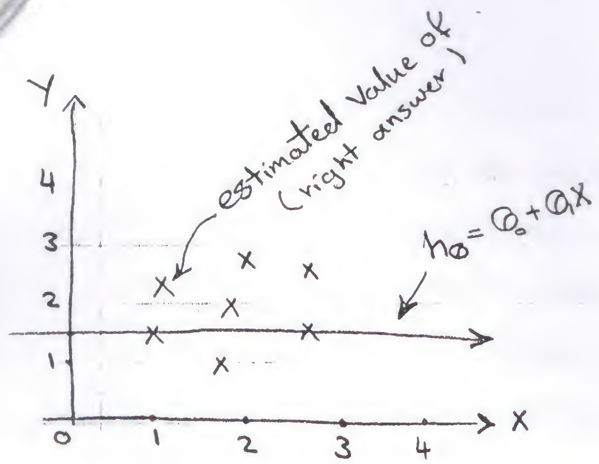
Regression

Hypothesis $h_\theta(x) = \theta_0 + \theta_1 x$

as we use straight Line as our model to fit data (x)

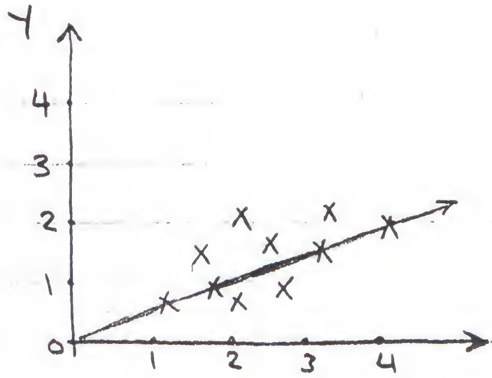
- we not only want to get the straight Line but we want to get the best one so we remain changing the values of θ_i (θ_0, θ_1) to get the best straight

∴ θ_0, θ_1 will let us figure out how to fit the best possible straight line to our data.



$\theta_0 = 1.5$

$\theta_1 = 0$



$\theta_0 = 0$

$\theta_1 = 0.5$

→ This figures show how the change of θ_0, θ_1 change $h_\theta(x)$

Idea: Choose θ_0, θ_1 so that $h_\theta(X)$ is close to y for our training examples (x, y).

→ X : input
Y : estimated output (Actual value of output)
h : approximate value of output

* we want to get difference between Actual value and approximate value (error) to know the quality of the straight line we get.

$$\therefore \text{error} = h_\theta(x^{(i)}) - y^{(i)}$$

we called error function by Cost Function. $J(\theta_0, \theta_1)$

- No to get -ve values we get square value and take avarage because we have number 'm' of training examples.

$$\therefore \underbrace{J(\theta_0, \theta_1)}_{\text{avarage mean square of the error}} = \frac{1}{2m} \sum_{i=1}^{m} \underbrace{(h_\theta(x^{(i)}) - y^{(i)})^2}_{\text{Error}}$$

mean square of the error

- We want to minimize $J(\theta_0, \theta_1)$ to get the best model fitting data.

We use MatLab to figure out Cost Function $J(\theta_0, \theta_1)$
we get 3D figure. in slide 18 lec 3

- Show slide 19, 20, 21, 22 in lec 3 to know that <u>for one line</u> θ_0, θ_1 change doesn't change the value of $J(\theta_0, \theta_1)$ as $J(\theta_0, \theta_1)$ is constant for the same line.

→ Cost Function determines the best model $h_\theta(x)$

→ Cost Function ≡ Objective Function $= J(\theta_0, \theta_1)$

- We want to fit data by the best model with minimam Cost Function
- We change the value of θ_0, θ_1 to get more models (straight lines) Fitting data we choose the best one depending on the error value or Cost Function , The value of θ_0, θ_1 of the best model is what we need.